IN THE

United States Court of Appeals

FOR THE DISTRICT OF COLUMBIA CIRCUIT

Nos. 00-5212, 5213

MICROSOFT CORPORATION,

Defendant-Appellant,

v.

UNITED STATES OF AMERICA and STATE OF NEW YORK, *et al*.,

Plaintiffs-Appellees.

ON APPEAL FROM THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF COLUMBIA

MICROSOFT CORPORATION'S PETITION FOR REHEARING

Pursuant to Rule 40 of the Federal Rules of Appellate Procedure and Circuit Rule 35, appellant Microsoft Corporation ("Microsoft") hereby petitions the Court to grant rehearing on a single narrow issue decided in the Court's June 28, 2001 Opinion. Microsoft respectfully submits that critical evidence was overlooked—or misinterpreted—on the technical question of whether Microsoft "commingled" software code specific to Web browsing with software code used for other purposes in certain files in Windows 98. *See* Op. at 38-39. The Court accepted the district court's conclusion that such "commingling" had occurred and that it violated Section 2 of the Sherman Act.

The Court's ruling with regard to "commingling" of software code is important because it might be read to suggest that OEMs should be given the option of removing

the software code in Windows 98 (if any) that is specific to Web browsing. *See* Op. at 38. The Government, however, did not seek such relief on appeal. *See* Appellees' Br. at 104. In fact, the Government made no effort to identify any software code specific to Web browsing in the same Windows 98 files as software code used for other purposes. The Government instead argued that removing end-user access to Internet Explorer was tantamount to removing Internet Explorer itself. *See id.* at 103-04.

Removal of end-user access by OEMs can be accomplished by including Internet Explorer in the Add/Remove Programs utility, which the Court concluded Microsoft should be required to do in any event. *See* Op. at 37. To the extent OEMs were ever deterred from installing third-party Web browsers on their machines because IE icons were present on the Windows desktop and in the Start menu, *see* Op. at 36-37, that problem will be fully addressed by including Internet Explorer in the Add/Remove Programs utility, which Microsoft has already announced it will do in response to the Court's decision.

Even if the Government could identify software code specific to Web browsing in the same Windows 98 files as software code used for other purposes, any notion that OEMs should be allowed to remove such software code—as opposed to removing end-user access to Internet Explorer—runs counter to the district court's finding that Microsoft's inclusion of Web browsing functionality in Windows benefits consumers. 84 F. Supp. 2d at 55, 110-11 (FF 186, 408). Furthermore, allowing OEMs to remove from Windows 98 the software code that supplies Web browsing functionality would be inconsistent with this Court's conclusion that Section 2 does not prohibit Microsoft from designing its operating systems so that Internet Explorer is automatically invoked to

support operating system features such as Windows Explorer and Windows Update. *See* Op. at 40. For example, if software code that supplies Web browsing functionality were removed from Windows 98, then Windows Explorer would no longer allow users to "move seamlessly from local storage devices to the Web *in the same browsing window*." *See* Op. at 40 (quoting Appellant's Opening Br. at 82) (emphasis in original). Similarly, users could not automatically upgrade their copy of Windows 98 using Windows Update.

In any case, as explained below, a detailed analysis of the factual record discloses that the district court's finding that Microsoft engaged in unjustified "commingling" of software code is clearly erroneous.

A.

"The purpose of a petition for rehearing is to ensure that the [Court] properly considers all relevant information when rendering the decision." 20A JAMES WM. MOORE ET AL., MOORE'S FEDERAL PRACTICE § 340.12, at 340-7 (3d ed. 2001). A petition for rehearing is appropriate where, as here, the Court may have overlooked or misapprehended facts of decisional significance. *See Moldea* v. *New York Times Co.*, 22 F.3d 310, 311 (D.C. Cir.), *cert. denied*, 513 U.S. 875 (1994); *see also United States* v. *Fields*, 251 F.3d 1041, 1043 (D.C. Cir. 2001); *Mancuso* v. *Herbert*, 166 F.3d 97, 99 (2d Cir.), *cert. denied*, 527 U.S. 1026 (1999); FED. R. APP. P. 40(a)(2).

The district court's condemnation of "commingling" appears in Finding of Fact 161. Specifically, the district court found that Microsoft improperly bound Internet Explorer to Windows by "placing code specific to Web browsing in the same files as code that provided operating system functions." 84 F. Supp. 2d at 50 (FF 161).

Of course, in general terms, "commingling of code" is not "per se pernicious or even suspicious." *United States* v. *Microsoft Corp.*, 147 F.3d 935, 958 (D.C. Cir. 1998) (Wald, J., concurring in part and dissenting in part); *see also id*. at 949 n.12. Indeed, "commingling" in the sense of having the same software code perform multiple functions in different computing contexts is a salutary consequence of technical integration. As the senior executive responsible for designing Windows 98, James Allchin, explained: "'The very same code in Windows 98 that provides Web browsing functionality' also performs essential operating system functions—not code in the same files, *but the very same software code*." Appellant's Opening Br. at 79 (emphasis in original) (quoting Allchin ¶ 9) (J.A. 3291-92). For example, it is undisputed that many of the same functions in the core Internet Explorer file called SHDOCVW.DLL are invoked when a user (i) views the contents of his or her local hard drive and (ii) displays a Web page. Such sharing of functions eliminates redundancy, reduces complexity and improves the performance of the operating system. *See* Allchin ¶ 53 (J.A. 3308); 2/3/99 am Tr. at 23-25 (Allchin) (J.A. 8105-07).

The Government concedes that it was permissible for Microsoft to employ "HTML rendering and other technologies used in browsing to provide non-browsing functions" in Windows 98, such as the new user interface, the Windows Update feature and the HTML Help system. Appellees' Br. at 66; *see also* Op. at 40. The Government also concedes that "IE expose[s] APIs" on which ISVs rely in developing applications to run on Windows. Appellees' Br. at 122; *see also* Op. at 83. Such reliance on Internet Explorer would not be possible if OEMs were able to remove it from the operating system.

As used by the district court in Finding of Fact 161, the term "commingling" does not refer to use of the same software code for multiple purposes. Rather, as noted above, "commingling" in the district court's parlance refers to placing software code specific to Web browsing together with software code used for other purposes in the same operating system files with no technical justification for doing so. Microsoft challenged Finding 161 as clearly erroneous because there is no "commingling" of software code in Windows 98 under the district court's definition.

This Court rejected Microsoft's challenge to Finding of Fact 161, identifying three putative bases for it. First, the Court noted that "[a] Government expert, Glenn Weadock, testified that Microsoft 'design[ed] [IE] so that some of the code that it uses co-resides in the same library files as other code needed for Windows.'" Op. at 38 (quoting Weadock ¶ 30). Second, the Court noted that another Government expert, Edward Felten, testified that one of the core Internet Explorer files in Windows, called SHDOCVW.DLL, "'contains some functions that have to do specifically with Web browsing, and it contains some general user interface functions as well.'" *Id*. at 38-39 (quoting 12/14/98 am Tr. at 60-61) (J.A. 6953-54). Third, the Court noted that one Microsoft document "suggests" that Microsoft may have "commingled" software code specific to Web browsing in the same operating system files with software code used for other purposes. *Id*. at 39 (citing GX 1686 (under seal)). None of these three bases can withstand scrutiny.

B.

1. Weadock's Testimony. The first piece of evidence cited by the Court, Glenn Weadock's testimony, provides no support whatsoever for Finding of Fact 161. On

cross-examination, Weadock admitted that he had never seen the source code (the human-readable instructions that get compiled into the object code that is directly readable by a computer) for Windows 98. 11/16/98 am Tr. at 37 (J.A. 5798). He also conceded that because he had no knowledge of the programming language C in which Microsoft's operating systems are written, he would not have understood the source code for Windows 98 even if he had seen it. *Id*. at 36-37 (J.A. 5797-98). In fact, Weadock freely acknowledged that he had no expertise in the design of operating systems:

> Q. You do not, however, consider yourself, sir, to be an expert in the design of operating system software, do you?
>
> A. No, I do not.

Id. at 35 (J.A. 5796). And another of the Government's experts, Edward Felten, stated: "I should point out to you that Mr. Weadock was not appearing as an expert in software design." 12/14/98 am Tr. at 68 (J.A. 6961).

Because Weadock knew nothing about the internal design of Windows 98, he had no basis to opine about how software code is organized into particular files in the operating system. His testimony on this subject is pure speculation and entitled to no weight. *See* 11/16/98 pm Tr. at 38-40 (J.A. 5886-88). Indeed, when specifically asked on cross-examination whether he was "suggesting that in designing Windows 98, Microsoft ha[d] commingled code units with other unrelated code units in a single DLL file," Weadock replied, "No, I'm not." 11/16/98 am Tr. at 69 (J.A. 5830); *see also id*. at 69-70 (J.A. 5830-31). He further admitted:

> Q. Just to be clear, it's not your testimony here today that there is no technical justification for the way the software code has been combined in Windows 98?

A. I'm not qualified to make that statement, and I do not make that statement.

11/16/98 pm Tr. at 40 (J.A. 5888).

What is more, when read in its entirety, even the sentence of Weadock's direct testimony quoted by the Court provides no independent support for Finding of Fact 161. The sentence states: "*As is discussed at greater length in the testimony of Dr. Edward Felten*, Microsoft, for example, has chosen to design Internet Explorer so that some of the code that it uses co-resides in the same library files as other code needed for Windows 98 or even Windows 95 to run." Weadock ¶ 30 (emphasis added). The sentence makes clear that Weadock, having no knowledge himself about the internal design of Windows 98, was relying entirely on Felten's views as the basis for his assertion. *See also* 11/16/98 pm Tr. at 39 (J.A. 5887) ("[T]here are other witnesses that will appear in this case that are highly qualified in those areas and more knowledgeable than I am."). When cross-examined on this sentence, Weadock again admitted that he had never seen the source code for Windows 98, thus revealing that he had no basis to assert that unidentified software code specific to Web browsing "co-resides" in the same files with software code used for other purposes. *Id.* at 38-39 (J.A. 5886-87).

2. Felten's Testimony. Edward Felten's testimony likewise provides no support for Finding of Fact 161. Despite his bald assertion that "[t]here is other code specific to IE Web browsing that could be deleted" from certain files in Windows, Felten expressly stated in his written direct testimony that he had made no attempt to identify such software code. Felten ¶ 58 (J.A. 3025). On cross-examination, Felten conceded that he had done no "study of what code might be removable or not removable in those files, because

. . . it's just not relevant to the questions that I was addressing." 12/14/98 am Tr. at 65 (J.A. 6958). Felten instead stated: "*If* there's code that's in there that can be removed, then certainly Microsoft is able to remove it." *Id*. (emphasis added). When asked whether he had made any effort to identify software code "solely related to Web browsing," Felten replied, "No, it just didn't seem—it just didn't seem like an important thing to consider . . . ." 6/10/99 pm Tr. at 46-47 (J.A. 9308-09). It was unimportant because the Government took the position at trial and on appeal that "[a]dding or removing the means of *user access* to a given function, by whatever means, amounts to adding or removing the software product," even if no software code is removed. Appellees' Br. at 104 (emphasis added).

Felten made no effort to determine whether there is any software code specific to Web browsing in SHDOCVW.DLL (or any other file in Windows 98 that includes software code used for other purposes) that can be removed from the operating system without impairing its functionality. His failure to do so—despite having the Windows 98 source code in his possession for more than nine months—deprives his testimony of any probative weight. Indeed, in the more than three years that this case has been pending, the Government has never identified any software code in Windows 98 that meets this description. Rather, as noted above, the Government focused its attention on removing end-user access to Internet Explorer.

3. GX 1686. The Microsoft document cited by the Court, GX 1686, likewise provides no basis to sustain Finding of Fact 161. James Allchin of Microsoft—who has a doctorate in computer science—was the only witness to testify about GX 1686, a chart prepared by a Windows developer named David D'Souza (who was not himself deposed

in the case). *See* 2/2/99 am Tr. at 68-76; 2/2/99 pm Tr. at 34-38 (J.A. 8040-44). Allchin explained that D'Souza had examined which functions in SHDOCVW.DLL were called when Internet Explorer was used in two specific scenarios: (i) opening the My Computer window to look at the contents of the computer's hard disk, and (ii) browsing Microsoft's Web site on the Internet. 2/2/99 pm Tr. at 34-35 (J.A. 8040-41). Looking only at those two scenarios, D'Souza determined that 1,061 out of 1,769 functions were shared, approximately 60 percent. Although D'Souza observed that his limited inquiry could be misinterpreted as supporting Felten's view that SHDOCVW.DLL was capable of being broken into two pieces, he concluded: "Clearly the integration is good. There is a HUGE amount of sharing and commonality here." GX 1619 (emphasis in original).

Allchin explained that additional functions in SHDOCVW.DLL would be invoked during normal use of a computer (beyond the two scenarios D'Souza examined), causing the number of shared functions to increase substantially:

> If we'd have taken a system and we were to run that system over a period of days and continued to look at it—even assuming just one DLL, SHDOCVW—and we had looked at some other Web sites and we had looked at a lot more files on the local machine with a lot more richness there, the [amount] of sharing[] that you would have had would have gone up.

2/2/99 am Tr. at 69. For example, the number of shared functions in SHDOCVW.DLL would increase if a user accessed (i) Web pages with complex scripts or multimedia content on the one hand and (ii) an operating system folder containing thumbnail views of PowerPoint presentations on the other hand—scenarios involving what Allchin referred to as "more richness."

With regard to D'Souza's analysis, Allchin stated: "You can't figure out anything from this other than that there is a lot of sharing, and if you did other scenarios there would be more sharing . . . ." 2/2/99 pm Tr. at 37 (J.A. 8043); *see also* 2/2/99 am Tr. at 68-69 ("But he only did two scenarios. And the only thing you can prove from that test, which is why I didn't put it in my testimony, is because you can only prove sharing if you run enough scenarios."). Rather than supporting the suggestion that SHDOCVW.DLL is a random assemblage of "commingled" software code, GX 1686 confirms that the composition of the file was intended to achieve the efficient use of shared functions.

Contrary to Finding of Fact 161, Microsoft did not "commingle" software code specific to Web browsing with software code used for other purposes in the same files in Windows 98. Rather, in organizing software code into files, Microsoft placed related functions close to one another. That improved the performance of the operating system by eliminating the overhead inherent in copying large numbers of disparate files from the computer's hard disk into active memory in order to perform a given task. *See* 2/3/99 am Tr. at 23 (J.A. 8105) ("What we do in DLL's is we try to rearrange the files—the source files that are related to each other so they'll be close to each other. And we do that to have a better performance experience for the user.").

In fact, Allchin expressly denied that Microsoft ever "commingled" software code specific to Web browsing with software code used for other purposes:

> Q. Is it correct, Mr. Allchin, that Microsoft intentionally mixed up the code that performs Web browsing functionality with unrelated code in the operating system in an effort to make it difficult for someone to identify and remove that code?

> A. That is absolutely untrue. We never did that. And it would not make good engineering sense to do that. We put the pieces together that we thought made sense.

2/3/99 am Tr. at 25-26 (J.A. 8107-08). The Government, which had the burden of proof, never refuted this testimony with competent evidence.

Based on the foregoing, it is plain that Finding of Fact 161 is clearly erroneous. As such, it cannot provide a basis for the Court's ruling that Microsoft violated Section 2 by "commingling" software code specific to Web browsing with software code used for other purposes in the same files in Windows 98. The district court's interpretation of the evidence is not plausible in light of the record viewed in its entirety. *See* Op. at 39 (citing *Anderson* v. *City of Bessemer City*, 470 U.S. 564, 573-74 (1985)).

C.

As this Court has observed, "even if there is some evidence supporting a finding, that finding is clearly erroneous if 'on the entire evidence [the reviewing court] is left with the definite and firm conviction that a mistake has been committed.'" *Cuddy* v. *Carmen*, 762 F.2d 119, 124 (D.C. Cir.) (quoting *United States* v. *U.S. Gypsum Co.*, 333 U.S. 364, 395 (1948)), *cert. denied*, 474 U.S. 1034 (1985). A review of the entire record reveals that the only support for Finding of Fact 161 is (i) the uninformed speculation of Weadock who never saw the Windows source code and admitted that he has no expertise in operating system design, (ii) the *ipse dixit* of Felten who chose to perform no analysis to support his conclusion and never identified any "commingled" software code in Windows 98, and (iii) a chart created by a Windows developer who examined one Internet Explorer file in two limited scenarios and concluded that the software code in that file was used extensively for both Web browsing and other purposes. Such attenu-

ated evidence cannot overcome Allchin's flat denial that Microsoft mixed software code specific to Web browsing code with software code used for other purposes in order to make it difficult to identify and remove such software code from Windows 98.

The absence of probative evidence supporting Finding of Fact 161 should leave the Court with the definite and firm conviction that a mistake has been made. In light of that mistake, there is no basis in the record for finding that Microsoft's product design violated Section 2 by reason of unjustified "commingling" of software code.

Accordingly, Microsoft respectfully requests that the Court grant rehearing on the question of whether Microsoft violated Section 2 of the Sherman Act by "commingling" software code specific to Web browsing with software code used for other purposes in the same files in Windows.

Respectfully submitted,

_____

William H. Neukom
Thomas W. Burt
David A. Heiner, Jr.
MICROSOFT CORPORATION
One Microsoft Way
Redmond, Washington 98052
(425) 936-8080

John L. Warden
Richard J. Urowsky
Steven L. Holley
Richard C. Pepperman, II
SULLIVAN & CROMWELL
125 Broad Street
New York, New York 10004
(212) 558-4000

Counsel for Defendant-Appellant Microsoft Corporation

July 18, 2001

CERTIFICATE OF SERVICE

I hereby certify that on this 18th day of July, 2001, I caused two true and correct copies of the foregoing Microsoft Corporation's Petition for Rehearing to be served by hand upon:

Phillip R. Malone
Antitrust Division
U.S. Department of Justice
325 Seventh Street, N.W.
Suite 615
Washington, D.C. 20530
Tel: (415) 436-6675
Fax: (415) 436-6687
Counsel for Appellee United States

Catherine G. O'Sullivan
Chief, Appellate Section
U.S. Department of Justice
Antitrust Division
601 D Street, N.W.
Room 10536
Washington, D.C. 20530
Tel: (202) 305-7420
Fax: (202) 514-0536
Counsel for Appellee United States

Louis R. Cohen
C. Boyden Gray
Wilmer, Cutler & Pickering
2445 M Street, N.W.
Washington, D.C. 20037-1420
Tel: (202) 663-6000
Fax: (202) 663-6363
Counsel for ACT and CompTIA

Paul T. Cappuccio
Randall J. Boe
Theodore W. Ullyot
America Online, Inc.
22000 AOL Way
Dulles, Virginia 20166
Tel: (703) 448-8700
Fax: (703) 265-1495
Counsel for AOL

Edward J. Black
Jason M. Mahler
Computer & Communications
Industry Association
666 Eleventh Street, N.W.
Washington, D.C. 20001
Tel: (202) 783-0070
Fax: (202) 783-0534
Counsel for CCIA

David R. Burton, Esq.
333 N. Fairfax Street
Suite 302
Alexandria, Virginia 22314-2632
Tel: (703) 548-5868
Fax: (703) 548-5869
Counsel for CMDC

Donald M. Falk
Mayer, Brown & Platt
1909 K Street, N.W.
Washington, D.C. 20001
Tel: (202) 263-3000
Fax: (202) 263-3300
Counsel for SIIA

Carl Lundgren
Valmarpro Antitrust
5035 South 25th Street
Arlington, Virginia 22206-1057
Tel: (703) 235-1910
Fax: (703) 235-5551

Robert H. Bork
1150 17th Street, N.W.
Washington, D.C. 20036
Tel: (202) 862-5851
Fax: (202) 862-5899
Counsel for ProComp

By facsimile and overnight courier upon:

Richard L. Schwartz
Deputy Chief, Antitrust Bureau
New York State Attorney General's Office
120 Broadway, Suite 2601
New York, New York 10271
Tel: (212) 416-8284
Fax: (212) 416-6015
Counsel for Appellee States

Christine Rosso
Chief, Antitrust Bureau
Illinois Attorney General's Office
100 West Randolph Street, 13th Floor
Chicago, Illinois 60601
Tel: (312) 814-2503
Fax: (312) 814-2549
Counsel for Appellee States

Kevin J. O'Connor
Office of the Attorney General of Wisconsin
P.O. Box 7857
123 West Washington Avenue
Madison, Wisconsin 53703-7957
Tel: (608) 266-1221
Fax: (608) 267-2223
Counsel for Appellee States

Lee A. Hollaar
School of Computing
University of Utah
3190 Merrill Engineering Building
50 South Central Campus Drive
Salt Lake City, Utah 84112-9205
Tel: (801) 581-3203
Fax: (801) 581-5843

Robert S. Getman
359 West 29th Street
Suite G
New York, New York 10001
Tel: (212) 594-6721
Fax: (212) 594-6732
Counsel for TAFOL

_____
Bradley P. Smith